Exploring Binary Classification Hidden within Partial Label Learning

Hengheng Luo[1] , **Yabin Zhang**[2] , **Suyun Zhao**[1*] , **Hong Chen**[1] and **Cuiping Li**[1]
[1]Key Lab of Data Engineering and Knowledge Engineering of MOE Renmin University of China
[2]Gaoling School of Artificial Intelligence, Renmin University of China, Beijing 100872, China
{luohengheng, yabin.zhang, zhaosuyun, chong}@ruc.edu.cn, cuiping_li@263.net

Abstract

Partial label learning (PLL) is to learn a discriminative model under incomplete supervision, where each instance is annotated with a candidate label set. The basic principle of PLL is that the unknown correct label y of an instance $\boldsymbol{x}$ resides in its candidate label set s, i.e., $P(y \in s|\boldsymbol{x}) = 1$. On which basis, current researches either directly model $P(y|\boldsymbol{x})$ under different data generation assumptions or propose various surrogate multiclass losses, which all aim to encourage the model-based $P_\theta(y \in s|\boldsymbol{x}) \to 1$ implicitly. In this work, instead, we explicitly construct a binary classification task toward $P(y \in s|\boldsymbol{x})$ based on the discriminative model, that is to predict whether the model-output label of $\boldsymbol{x}$ is one of its candidate labels. We formulate a novel risk estimator with estimation error bound for the proposed PLL binary classification risk. By applying logit adjustment based on disambiguation strategy, the practical approach directly maximizes $P_\theta(y \in s|\boldsymbol{x})$ while implicitly disambiguating the correct one from candidate labels simultaneously. Thorough experiments validate that the proposed approach achieves competitive performance against the state-of-the-art PLL methods.

1 Introduction

Trained with precisely annotated data, recent applications of machine learning have achieved extraordinary success in various real-world scenarios. However, collecting large datasets with high-quality annotation is expensive and almost unrealistic. In comparison, it is more feasible to learn under incomplete supervision. In this paper, we consider an important weakly supervised setting called partial label learning (PLL), where each instance is annotated with a set of candidate labels containing the ground-truth label, while other irrelevant labels are treat as non-candidate labels. The original label space is divided coarsely into the positive part containing the candidate labels and the negative part containing the non-candidate labels. PLL has been successfully applied to different application domains [Liu and Dietterich, 2012; Zeng *et al.*, 2013] and it has attracted considerable attention of researchers, especially when generalizing to the modern deep learning [Yan and Guo, 2020; Seo and Huh, 2021].

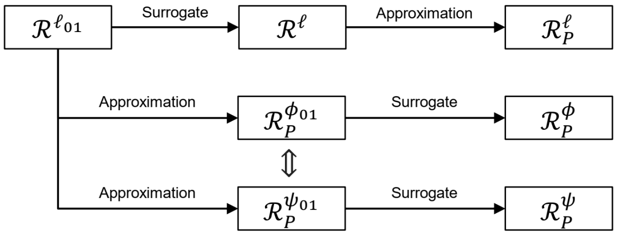


Figure 1: The learning steps of PLL in different orders. The Surrogate step represents replacing 0-1 loss with a surrogate loss, and the Approximation step represents modifying the risk from standard distribution to partial distribution.

The pioneering work proposed to minimize the KL divergence between the model-based distribution and the given class prior [Jin and Ghahramani, 2002]. Following this, numerous studies proposed to add some constraints to the objective function, and the model parameters are optimized by utilizing EM algorithm [Feng and An, 2018; Feng and An, 2019]. Besides, some previous studies focused on adapting widely-used learning techniques to PLL, such as maximum margin [Nguyen and Caruana, 2008], k-nearest neighbors [Hüllermeier and Beringer, 2006] and so on. However, the optimization-constrained objectives of these classical methods can not be compatible with deep neural networks.

The focus of recent studies in PLL is summarized in Figure 1. Researchers define the partial 0-1 risk $\mathcal{R}_P^{\phi_{01}}$ to upper-bound the 0-1 risk $\mathcal{R}^{\ell_{01}}$ [Cour *et al.*, 2011] and minimize the partial surrogate risk $\mathcal{R}_P^{\phi}$ through the middle learning steps [Lv *et al.*, 2020; Seo and Huh, 2021]. Furthermore, by modeling the generative relationship between $P(y|\boldsymbol{x})$ and $P(s|\boldsymbol{x})$, the surrogate risk $\mathcal{R}^{\ell}$ can be reformulated as $\mathcal{R}_P^{\ell}$ for the partial distribution through the top learning steps [Feng *et al.*, 2020; Wen *et al.*, 2021]. In summary, most researchers have been working on training a discriminative model toward $P(y|\boldsymbol{x})$ following the multiclass-based risk minimization principle, which seems the only promising solution to solve PLL problem. One related research [Liu and Dietterich, 2014] proposed the little-noticed notion of binary classifica-

[*]Corresponding author

tion task derived from the PLL task, which provides us an alternative through the bottom learning steps, namely, PLL binary classification (PLL-BC) risk $\mathcal{R}_P^{\psi}$ estimator. In this manner, a binary classifier is induced from the original multiclass classifier to estimate whether the predicted label of an instance is inside of its candidate label set, which is explicitly modeling $P(y \in s|\boldsymbol{x})$. However, limited theoretical analysis is not comprehensive since it naturally raises an equally important question: how to design a surrogate PLL-BC loss to recover the solution of the original multiclass problem under the guidance of theoretical results?

This paper gives a positive answer to this question through a theoretical and empirical analysis. The basic principle of PLL highly motivates us to directly estimate $P(y \in s|\boldsymbol{x})$ from the discriminative model's output, which naturally leads to the binary classification task. Accordingly, we first investigate the feasibility of the proposed PLL-BC task based on 0-1 loss and then derive a novel risk estimator with estimation error bound, which means learning in the context of partial labels is consistent and practical. Different from existing multiclass loss approaches, the proposed PLL-BC loss is formulated by logit adjustment based on classical disambiguation strategy, i.e., identification-based strategy. Concretely, a useful weight normalization strategy on logits is introduced to dynamically reassign label weights during each training iteration, which encourages the model to implicitly identify the correct label. Finally, experimental results on a variety of datasets show that our approach achieves competitive performance against the state-of-the-art approaches.

2 Background

This section reviews the formulations of multiclass classification and partial label learning, and we briefly introduce some recent developments for each.

2.1 Multiclass Classification

Given $\boldsymbol{x} \in \mathbb{R}^d$ (d is the dimensionality) as the input random variable in the feature space $\mathcal{X}$ and $y \in \mathbb{R}$ as the output random variable in the label space $\mathcal{Y} = [k]$ (with k classes) where $[k] := \{1, 2, \ldots, k\}$, we assume that each example $(\boldsymbol{x}, y)$ is sampled independently and identically from an unknown joint data distribution over $\mathcal{X} \times \mathcal{Y}$ with probability density $p(\boldsymbol{x}, y)$. The task of multiclass classification is to learn a classifier $\boldsymbol{g} : \mathcal{X} \rightarrow \mathbb{R}^k$. Formally, we denote $g_i(\boldsymbol{x})$ is the estimate of $P(y = i|\boldsymbol{x})$ and $\sum_{i=1}^{k} g_i(\boldsymbol{x}) = 1$. The 0-1 risk is typically of the following form:

$$\mathcal{R}^{\ell_{01}}(\boldsymbol{g}) = \mathbb{E}_{p(\boldsymbol{x},y)} \left[\ell_{01}\left(\boldsymbol{g}(\boldsymbol{x}), \boldsymbol{e}^y\right)\right] \tag{1}$$

where $\ell_{01} = \mathbb{I}\left(\arg\max_{i \in \mathcal{Y}} \boldsymbol{g}_i(\boldsymbol{x}) \neq y\right)$ is the 0-1 loss (standard classification error), and $\boldsymbol{e}^y \in \{0, 1\}^k$ is the one-hot vector, i.e., the y-element in $\boldsymbol{e}^y$ is one and others are zero. Besides, one surrogate expression of Eq. (1) is defined as

$$\mathcal{R}^{\ell}(\boldsymbol{g}) = \mathbb{E}_{p(\boldsymbol{x},y)} \left[\sum_{i=1}^{k} \ell\left(g_i(\boldsymbol{x}), e_i^y\right)\right], \tag{2}$$

where e_i^Y is the i-th element of e^Y and ℓ is some decomposable continuous non-negative loss.

2.2 Partial Label Learning

Different from traditional explicit supervision, partial label is a kind of ambiguous label information. Each instance is given a set of candidate labels, among which only one is the ground truth label while others are false positives. Given a partially labeled dataset $\{(\boldsymbol{x}_i, s_i)\}_{i=1}^{n}$, where the candidate labels set $(s_i)_{i \leq n} \in \mathcal{S}$ and $\mathcal{S} = \left\{2^{\mathcal{Y}} \backslash \emptyset \backslash \mathcal{Y}\right\}$ is the space of closed subsets of $\mathcal{Y}$, the target of PLL is to learn a discriminative model which identifies the ground-truth label of unseen data. Formally, we assume that each random variable $(\boldsymbol{x}, s)$ is drawn from an unknown data distribution with the margin density $p(\boldsymbol{x}, s)$ of $p(\boldsymbol{x}, y, s)$, and the ground-truth label y for an instance $(\boldsymbol{x}, s)$ is not directly accessible during training phase but exists in the candidate label set, which is defined as $P(y \in s|\boldsymbol{x}) = 1$. The partial 0-1 risk is defined as

$$\mathcal{R}_P^{\phi_{01}}(\boldsymbol{g}) = \mathbb{E}_{p(\boldsymbol{x},s)} \left[\phi_{01}\left(\boldsymbol{g}(\boldsymbol{x}), s\right)\right], \tag{3}$$

where $\phi_{01} = \mathbb{I}(\arg\max_{i \in \mathcal{Y}} g_i(\boldsymbol{x}) \notin s)$ is the partial 0-1 loss (partial classification error). In front of big data, the above-mentioned classical studies may be inefficient because of the high time complexity and objectives optimization problems. Hence, many deep PLL methods have been proposed. [Zhang *et al.*, 2020; Seo and Huh, 2021].

Note that the ECOC-based algorithm proposed by [Zhang *et al.*, 2017] and the One-vs-One-based algorithm proposed by [Wu and Zhang, 2018] are both binary decomposition methods for inducing the predictive model while our work focuses on training a discriminative model by minimizing surrogate PLL *binary classification* risk. Another related study is the surrogate complementary loss framework [Chou *et al.*, 2020], which focuses on the negative risk and overfitting effects derived from Complementary Label Learning problem while our work is derived in a totally different manner.

3 Methodology

In this section, we first demonstrate the feasibility analysis of the proposed PLL-BC task, and then represent a simple risk estimator for the PLL-BC risk. Next, we introduce logit adjustment strategy to design the PLL-BC loss and theoretically derive an estimation error bound for the proposed method.

3.1 Binary Classification Task of PLL

We start with the PLL-BC task by constructing a binary classification task from the PLL setting. Following [Liu and Dietterich, 2014], the binary classifier $\boldsymbol{f} : \mathcal{X} \rightarrow \mathbb{R}^k \rightarrow \mathbb{R}^2$ builds on the standard multiclass classifier $\boldsymbol{g}$, which is denoted as

$$\boldsymbol{f}(\boldsymbol{x}, s) = [f_0(\boldsymbol{x}, \overline{s}), f_1(\boldsymbol{x}, s)], \tag{4}$$

where $\overline{s}$ is the non-candidate label set, f_1 is manually treat as the estimate of $P(y \in s|\boldsymbol{x})$ and $f_0 + f_1 = 1$. The reconstructed label space is defined as $\tilde{y} = [0, 1]$. With only accessing the ambiguous supervision, the next-best option of $\boldsymbol{f}$ is to estimate whether the predicted label from $\boldsymbol{g}$ is inside of the candidate label set, i.e., $f_1 = P(\arg\max_{i \in \mathcal{Y}} g_i(\boldsymbol{x}) \in s)$. Semantically, this inducing relation is a surjection from the original multiclass classification hypothesis space $\mathcal{G}$ to the binary classification hypothesis space $\mathcal{F}$. Then, the PLL binary classification error is exactly the partial classification error, i.e., $\mathcal{R}_P^{\phi_{01}}(\boldsymbol{g}) \Leftrightarrow \mathcal{R}_P^{\psi_{01}}(\boldsymbol{f})$, as shown in Figure 1. Next, we revisit the theoretical study on the learnability of PLL. The ambiguity degree is defined as

$$\gamma = \sup_{(\boldsymbol{x},y)\sim p(\boldsymbol{x},y), \bar{y}\in\mathcal{Y}, s\sim p(s|\boldsymbol{x},y), \bar{y}\neq y} P(\bar{y} \in s), \tag{5}$$

which represents the maximum probability of a negative label $\bar{y}$ co-occurs with the correct label y. Then, we can derive the learnability of PLL relates to the ambiguity degree.

Proposition 1. *(Summarized from [Cour* et al.*, 2011; Liu and Dietterich, 2014]) When the ambiguity degree satisfies* $0 \leq \gamma < 1$*, for the multiclass classifier* $\boldsymbol{g}$ *and the proposed binary classifier* $\boldsymbol{f}$*, the following comparison inequality holds:*

$$\mathcal{R}_P^{\psi_{01}}(\boldsymbol{f}) \leq \mathcal{R}^{\ell_{01}}(\boldsymbol{g}) \leq \frac{1}{(1-\gamma)} \mathcal{R}_P^{\psi_{01}}(\boldsymbol{f}).$$

This proposition allows us to bound the risk of the original multiclass problem with the risk measured with the partial loss, that is to approximately minimize the standard loss with access only to the PLL-BC one.

3.2 PLL Binary Classification Risk Estimator

Correspondingly, we propose the PLL-BC loss trained with DNNs. Formally, we learn a deep model by parametrizing the embeddings $\boldsymbol{h}(\boldsymbol{x};\theta) : \mathcal{X} \rightarrow \mathbb{R}^k$ of a DNN, which maps each data point $\boldsymbol{x}$ to k real-valued numbers known as *output logits*. Traditionally, these output logits are used to parameterize a predictive categorical distribution $p_\theta(y|\boldsymbol{x})$ by using the softmax function. In standard discriminative setting, the label with maximum logit is selected as the predicted label, i.e., $\arg\max_{i\in\mathcal{Y}} h_i(\boldsymbol{x};\theta)$. If an instance $(\boldsymbol{x}, s)$ is drawn from the complete distribution with density $p(\boldsymbol{x}, y, s)$, then $P(y \in s|\boldsymbol{x}) = 1$ holds. According to this basic setting for PLL, the virtual label $e_1^{\tilde{y}}$ is always the ground-truth prediction for $\boldsymbol{f}$. The optimization direction of the binary classifier is to increase the predictive gap between f_1 and f_0. Capturing this idea, the surrogate PLL-BC Loss is defined as

$$\psi(\boldsymbol{f}(\boldsymbol{x}, s), \boldsymbol{e}^{\tilde{y}}) = \tilde{\ell}(f_1(\boldsymbol{x}, s), e_1^{\tilde{y}}) + \tilde{\ell}(f_0(\boldsymbol{x}, \bar{s}), e_0^{\tilde{y}}), \tag{6}$$

where $\tilde{\ell}$ is similar with the standard loss ℓ, i.e., $\tilde{\ell}(\hat{\eta}, \eta) = 0$ only when $\hat{\eta} = \eta$. Note that the above equation is derived from the standard binary classification loss while the only difference is that $\boldsymbol{f}$ is induced from the original multiclass model. The novel risk estimator for PLL-BC is defined as

$$\mathcal{R}_P^{\psi}(f) = \mathbb{E}_{p(\boldsymbol{x},s)}\left[\psi(\boldsymbol{f}(\boldsymbol{x}, s), \boldsymbol{e}^{\tilde{y}})\right], \tag{7}$$

and the empirical risk estimator is rewritten as follows:

$$\widehat{\mathcal{R}}_P^{\psi}(f) = \frac{1}{n}\sum_{i=1}^{n}\left[\psi(\boldsymbol{f}(\boldsymbol{x}_i, s_i), \boldsymbol{e}^{\tilde{y}})\right]. \tag{8}$$

3.3 Predictive Probability via Logit Adjustment

Next, we propose logit adjustment strategy to induce the binary classifier toward $P(y \in s|\boldsymbol{x})$. Commonly, the output logits can be treat as an intuitive metric to measure the label confidence, which means the output logit regarding label i is positively correlated to its model-based conditional probability: $h_i(\boldsymbol{x};\theta) \propto P_\theta(y = i|\boldsymbol{x})$ [Grathwohl *et al.*, 2020]. Following the classical disambiguation strategies, we propose two baselines and a generalization of them.

Average-based Strategy

Based on the principle of maximum entropy, a natural assumption is that each candidate label makes equal contributions to the discriminative model. If the average logit of candidate labels is larger than that of non-candidates, a reasonable prospect is that the multiclass classifier tends to predict the potential correct label from candidates. We define

$$f_1(\boldsymbol{x}, s) = \frac{\exp\left(\frac{1}{|s|}\sum\limits_{i\in s} h_i(\boldsymbol{x})\right)}{\exp\left(\frac{1}{|s|}\sum\limits_{i\in s} h_i(\boldsymbol{x})\right) + \exp\left(\frac{1}{|\bar{s}|}\sum\limits_{j\in \bar{s}} h_j(\boldsymbol{x})\right)} \tag{9}$$

as the model-based positive probability output at the logit level, where $\boldsymbol{h}(\boldsymbol{x};\theta)$ is abbreviated to $\boldsymbol{h}(\boldsymbol{x})$. Note that f_0 can be formulated as $1 - f_1$. Intuitively, Eq. (9) is parametrized by a softmax normalization and the virtual positive logit is defined as the mean value of candidate logits. However, this strategy alone does not guarantee that the label regarding the maximum logit is one of the candidate labels, which goes against our desideratum.

Identification-based Strategy

Note that the task of PLL is to identify the label with the highest relevant degree. Ideally, the label confidence of the ground-truth label should be larger than others. Accordingly, the logits of other candidate labels do not need to be large, and that of the non-candidate labels should be as small as possible. Thus, our key observation in this work is

$$\max_{i\in s} P(y = i|\boldsymbol{x}) > \max_{j\in\bar{s}} P(y = j|\boldsymbol{x}). \tag{10}$$

This inequality is simple and explicable. Firstly, the conditional probability of the potential correct label should account for the largest proportion of the candidate label set. Secondly, each non-candidate label is definitely not the correct label, which means the label confidence of non-candidates should be considerably small. One may see that from the perspective of decision boundary, the margin-based algorithm proposed by [Nguyen and Caruana, 2008] can be treat as one specific implementation based on this inequality. Let us define

$$f_1(\boldsymbol{x}; s) = \frac{\exp\left(\max\limits_{i\in s} h_i(\boldsymbol{x})\right)}{\exp\left(\max\limits_{i\in s} h_i(\boldsymbol{x})\right) + \exp\left(\max\limits_{j\in\bar{s}} h_j(\boldsymbol{x})\right)}, \tag{11}$$

which focuses on only two most representative logits: the potential ground-truth logit and the semantically similar negative logit. Maximizing Eq. (11) will encourage the model to increase the predictive confidence of the potential ground-truth label while reducing that of non-candidate labels. With this aggressive selection, the original multiclass model will be forced to output the candidate label regarding maximum logit. Following this constructing idea, we establish the estimation error bound for the proposed method under mild assumptions.

To begin with, suppose $\tilde{\ell}(g_k(\boldsymbol{x}), e_k^y) = \ell(g_k(\boldsymbol{x}), e_k^y)$ satisfies $\tilde{\ell}(1 - g_k(\boldsymbol{x}), 1 - e_k^y) \leq \tilde{\ell}(g_k(\boldsymbol{x}), e_k^y)$ for each variable, i.e., MSE loss. The classifiers $\boldsymbol{g}$ and $\boldsymbol{f}$ are calculated through the same normalization function, i.e., softmax function. Let the Rademacher complexity of $\mathcal{G}$ over $p(\boldsymbol{x})$ with sample size n be defined as $\mathfrak{R}_n(\mathcal{G})$ [Mohri *et al.*, 2012]. Let $\boldsymbol{f}^*$ be the

Algorithm 1 PLL via Partial Binary Classification Loss

```
Input:
$\mathcal{D}$:   the partially labeled dataset $\mathcal{D} = \{(\boldsymbol{x}_i, s_i)\}_{i=1}^{n}$
Parameter:
$T$:   the number of epochs
$\lambda$:   the scale coefficient
Output:
$\theta$: the model parameters for $\boldsymbol{h}(\boldsymbol{x}; \theta)$
Initialize model parameters $\theta$;
for $t = 0$ to $T$ do
    Calculate original output logits by $\boldsymbol{h}(\boldsymbol{x}; \theta)$;
    Calculate $\boldsymbol{f}(\boldsymbol{x}, s)$ through logit adjustment strategy, i.e., Eq. (15);
    Compute empirical risk by Eq. (8);
    Update $\theta$ by optimizer;
end for
```

optimal classifier and $\hat{\boldsymbol{f}}$ be the empirical risk classifier. We use M and L_ℓ to represent the upper bound and Lipschitz constant of the original loss function ℓ respectively. Next, we derive the following estimation error bound.

Theorem 1. *For any $\delta > 0$, with probability at least $1 - \delta$,*

$$\mathcal{R}_P^{\psi}(\hat{\boldsymbol{f}}) - \mathcal{R}_P^{\psi}(\boldsymbol{f}^*) \leq 4L_\ell \mathfrak{R}_n(\mathcal{G}_k) + 2M\sqrt{\frac{\log(2/\delta)}{2n}}. \quad (12)$$

This theorem guarantees that in context of partial labels, learning with PLL-BC loss is consistent for all parametric models: as $n \to \infty$, $\mathfrak{R}_n(\mathcal{G}_k) \to 0$, then $\mathcal{R}_P^{\psi}(\hat{\boldsymbol{f}}) \to \mathcal{R}_P^{\psi}(\boldsymbol{f}^*)$ with convergence rate $\mathcal{O}(1/\sqrt{n})$.

Weight Normalization Strategy

Both disambiguation strategies are intuitively clear but still have their shortcomings. When handling the outliers, the average-based strategy is robust under mild assumptions on the domination of correct labels. However, the ground-truth label might be overwhelmed by the false-positive labels during model training. For the identification-based strategy, the progressive identification puts more emphasis on the potential correct logit, which is crucial for aligning with the correct one. However, the output logits of DNNs are sensitive, especially when the predictive information is inaccurate in the initial stage of model training, which leads the surrogate loss to be unreliable. To exploit the benefits of both, we replace the $\max$ operator of Eq. (11) with a weight normalization function. Consider a standard maximum function $\max(z_1, \ldots, z_k)$, we denote

$$\max_i z_i \approx \max_{\boldsymbol{q} \in \Delta} \sum_i q_i z_i, \quad (13)$$

where $\mathbf{q} \in \mathbb{R}^k$ is an unknown probability distribution, and $\Delta = \{\mathbf{q} | \sum_i q_i = 1, \forall i, q_i \geq 0\}$ denotes the simplex vector. In this way, the maximum logit is converted into a linear combination of k arguments. According to [Qian *et al.*, 2019], we obtain the closed form solution of the distribution $\boldsymbol{q}$ by adding the entropy regularizer to $\boldsymbol{q}$, which is defined as $q_i = \exp(\frac{1}{\lambda} z_i) / \sum_{j=1}^{k} \exp(\frac{1}{\lambda} z_j)$, where λ is a trade-off hyperparameter. Instead of assigning fixed values, updating label weights in each training epoch is a practical way to gradually approximate the ground-truth label weights [Lv *et al.*, 2020]. Specifically, for each candidate label $i \in s$, we define its corresponding label weight as

$$\mathrm{w}_i^{(t)} = \frac{\exp(\frac{1}{\lambda} h_i(\boldsymbol{x})^{(t)})}{\sum_{j \in s} \exp(\frac{1}{\lambda} h_j(\boldsymbol{x})^{(t)})}, \quad (14)$$

where t denotes the t-th epoch in model training. Hence, the positive predictive probability is defined as

$$f_1(\boldsymbol{x}, s) = \frac{\exp(\sum_{i \in s} w_i^{(t)} h_i(\boldsymbol{x}))}{\exp(\sum_{i \in s} w_i^{(t)} h_i(\boldsymbol{x})) + \exp(\sum_{j \in \bar{s}} w_j^{(t)} h_j(\boldsymbol{x}))}, \quad (15)$$

where $\sum_{j \in \bar{s}} w_j^{(t)} = 1$ and $\sum_{i \in s} w_i^{(t)} = 1$ respectively. This update strategy relaxes the $\max$ operator to a label weights combination. In particular, we treat the label weights as constant coefficients with respect to the parameters for backpropagation while manually updating their values during the training stage. When $\lambda \neq 1$, this can be seen as the temperature scaling applied to the label weights.

3.4 Learn with PLL-BC Loss

In implementation, the original output logits are calculated from the multiclass DNN's latent space embeddings and the binary output is calculated by Eq. (15). Then we adopt Eq. (6) as the empirical loss function to calculate the error information. The key steps of our method are outlined in Algorithm 1. After the completion of the model training process, the predicted label is given by $\arg\max_{i \in \mathcal{Y}} h_i(\boldsymbol{x})$.

4 Experiments

In this section, we verify the effectiveness of the proposed algorithm with extensive experiments on synthetic datasets and real-world datasets respectively. The best results among all methods are highlighted in bold and we use ● to represent that the proposed method is significantly better than the other baselines by using paired t-test at 5% significance level.

4.1 Datasets

We present experimental results on three widely-used benchmark datasets, i.e., MNIST [LeCun *et al.*, 1998], Fashion [Xiao *et al.*, 2017] and Kuzushiji [Clanuwat *et al.*, 2018]. We follow the problem settings in [Lv *et al.*, 2020] and the synthetic datasets are generated by a binomial flipping strategy or a pair flipping strategy. We construct a class transition matrix to artificially corrupt labels by a flipping probability q which denotes the probability to be selected as the candidate labels. In this paper, we consider $q \in \{0.2, 0.5\}$. By definition, we use only partially labeled data, while unlabeled data are ruled out. Furthermore, five real-world datasets are used from various application domains, including Lost [Cour *et al.*, 2009], MSRCv2 [Liu and Dietterich, 2012], BirdSong [Briggs *et al.*, 2012], Soccer Player [Zeng *et al.*, 2013] and Yahoo! News [Guillaumin *et al.*, 2010].

Data Generation: Binomial Flipping Strategy

Dataset	q	PLL-BC	RC	CC	PRODEN	DPNL	LWS
MNIST	0.2	**98.76**±**0.09**	98.66±0.05	98.56±0.05●	98.56±0.07●	98.66±0.02	98.52±0.02●
	0.5	**98.44**±**0.05**	98.33±0.06	98.27±0.07●	98.34±0.16	98.14±0.37	98.06±0.02●
Fashion	0.2	**89.84**±**0.23**	89.30±0.24●	88.78±0.47●	89.82±0.24	89.55±0.43	89.80±0.15
	0.5	**88.95**±**0.25**	88.53±0.06●	88.36±0.13●	88.87±0.13	88.42±0.30●	88.73±0.22
Kuzushiji	0.2	**93.43**±**0.12**	92.10±0.63●	91.72±0.36●	92.99±0.34●	92.73±0.39●	92.71±0.87
	0.5	**91.41**±**0.19**	91.20±0.26	90.81±0.18●	90.64±0.36●	90.63±1.25	90.82±0.18●

Data Generation: Pair Flipping Strategy

Dataset	q	PLL-BC	RC	CC	PRODEN	DPNL	LWS
MNIST	0.2	**98.78**±**0.09**	98.74±0.02	98.71±0.03	98.73±0.15	98.68±0.20	98.71±0.06
	0.5	**98.69**±**0.12**	98.44±0.08●	98.39±0.03●	98.65±0.12	98.61±0.14	98.64±0.10
Fashion	0.2	**90.48**±**0.16**	90.11±0.18●	90.16±0.14●	90.35±0.30	90.24±0.06●	90.36±0.19
	0.5	**90.27**±**0.16**	90.15±0.12	89.99±0.11●	90.10±0.31	90.13±0.28	90.26±0.12
Kuzushiji	0.2	**94.02**±**0.21**	93.42±0.12●	93.36±0.17●	93.59±0.47●	93.32±0.34●	93.96±0.29
	0.5	**93.87**±**0.22**	93.36±0.11●	92.91±0.15●	93.20±0.12●	93.33±0.45	93.69±0.26

Table 1: Test accuracy (mean±std) on the synthetic datasets (in %).

4.2 Baselines

For synthetic datasets, we compare the proposed method to five SOTA DNN based methods from PRODEN [Lv *et al.*, 2020], DNPL [Seo and Huh, 2021], LWS [Wen *et al.*, 2021] and RC&CC [Feng *et al.*, 2020]. For real-world datasets, we further compare our method with four classical PLL methods: IPAL [Zhang and Yu, 2015], PL-SVM [Nguyen and Caruana, 2008], PL-ECOC [Zhang *et al.*, 2017] and PL-KNN [Hüllermeier and Beringer, 2006]. We use PLL-BC to denote the proposed method where ψ is the binary cross entropy loss and $\boldsymbol{f}$ is constructed according to Eq. (15). For our method, the best hyperparameters are selected through grid search on a validation set, where learning rate $lr \in \{0.001, 0.005, 0.01, 0.05, 0.1\}$, weight decay $wd \in \{10^{-5}, \ldots, 10^{-3}\}$, and the temperature coefficient $\lambda \in \{0.5, \ldots 1.0\}$, with the learning rate decays halved per 50 epochs. For other methods, all hyper-parameters are searched according to the suggested parameter settings. We employ two base models, including linear model and 5-layer perceptron (MLP), and use SGD as the optimizer with a momentum of 0.9. The number of epoch is set as 250 and the mini-batch size is set as 256 for synthetic datasets and 128 for real-world datasets respectively. The implementations are based on PyTorch [Paszke *et al.*, 2019] and experiments are conducted with NVIDIA RTX 2080 Ti GPUs. More detailed descriptions of experiments are shown in Appendix.

4.3 Results on Synthetic Datasets

We report the mean values and standard error of test accuracy out of 5 trials. Table 1 reports the classification performance of each algorithm in the binomial case and pair case respectively. All deep PLL methods are trained with the same MLP model. Based on the results in Table 1, we clearly see that under the same base model, the proposed method achieves competitive performance against the state-of-the-art deep PLL approaches, showing that label disambiguation is accomplished with high quality on different synthetic datasets in different data generation cases. Collectively, the PLL-BC loss serves as an effective way for learning with partially labeled data, which matches its theory.

4.4 Results on Real-world Datasets

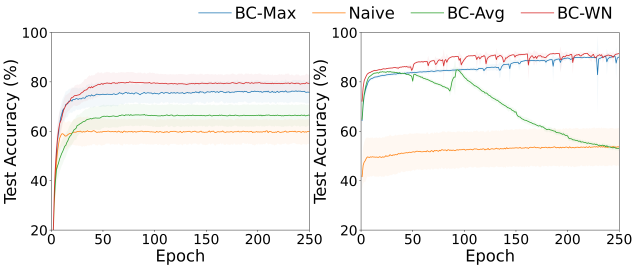
Means and standard deviations of each baseline are measured over 10-fold cross-validation, as shown in Table 2. For a fair comparison, all deep PLL methods employ a linear model. Compared to the deep PLL methods, PLL-BC loss overall achieves comparable or better performance on all datasets. Compared to the classical PLL methods, all deep PLL methods are at a competitive disadvantage on Birdsong and MSRCv2 datasets. It is reasonable that we adopt the linear model, and the representation ability of DNNs has not yet been fully exploited. Besides, we observe that there might be some extreme cases with large ambiguity degree in real-world datasets, then the proposed method would achieve mediocre performance. It is noteworthy that PLL-BC achieves superior performance against PL-KNN, where both methods are differently implemented but have similar functionality based on Eq. (10). We perform logit adjustment on the classifier to encourage a large relative margin between the output logits of candidates versus that of non-candidates while PL-KNN focuses on the minimum distance to the decision boundary. These results indicate the advantage of the logit margin-based formulation against other baselines.

4.5 Empirical Understandings

We conduct comprehensive experiments, on the key components introduced into the proposed approach. Specifically, experiments analysis is conducted on the Lost dataset and the Kuzushiji dataset in binomial case with $q = 0.5$.

	Lost	Birdsong	MSRCv2	Soccer Player	Yahoo! News
PLL-BC	**79.41±3.47**	71.97±2.06	44.66±4.87	**57.43±1.52**	**67.96±0.79**
PRODEN	76.11±4.63●	71.81±2.18	44.54±3.94	56.42±1.48●	67.40±0.73●
DNPL	75.67±3.40●	71.79±2.20	44.03±4.40	54.84±1.22●	67.27±1.85
RC	78.70±4.55	71.65±2.29	47.38±4.27	56.75±0.97	67.90±0.91
CC	77.72±4.13●	71.85±2.04	45.22±2.59	56.32±0.89●	67.91±0.71
LWS	78.97±3.74	65.07±2.06●	39.48±2.51●	51.56±1.71●	46.41±3.77●
PL-KNN	34.14±4.17●	64.37±2.13●	43.29±5.29●	49.23±1.46●	41.30±1.25●
PL-SVM	71.74±4.53●	45.40±3.58●	31.46±3.02●	38.56±4.37●	51.56±1.54●
PL-ECOC	67.38±4.27●	**74.15±1.94**	46.53±3.22	29.49±6.15●	61.81±1.09●
IPAL	73.08±4.37●	71.27±2.06●	**52.39±4.29**	55.01±1.01●	66.76±0.90●

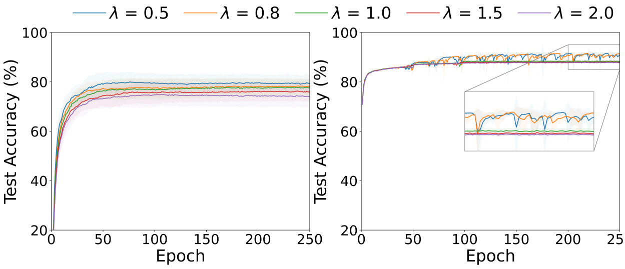
Table 2: Test accuracy (mean±std) on the real-world datasets (in %).



(a) Lost, Linear

(b) Kmnist, MLP

Figure 2: Study of the logit adjustment strategy.



(a) Lost, Linear

(b) Kmnist, MLP

Figure 3: Study of the temperature coefficient in PLL-BC loss.

Effectiveness Analysis of Logit Adjustment

By applying logit adjustment, the binary classifier is constructed from the original multiclass classifier. We compare the performance under different logit adjustment strategies. Furthermore, the multiclass-based minimal loss (Naive) is used as baseline, which simply selects the maximum logit as the correct one [Lv *et al.*, 2020]. We use BC-Avg, BC-Max and BC-WN to represent the PLL-BC loss implemented with Eq. (9), (11) and (15) respectively. As shown in Figure 2, we first observe that the BC-Max loss substantially outperforms the minimal loss and even achieves competitive performance against the BC-WN loss on Kuzushiji datasets. This phenomenon empirically verifies that by exploiting the current predictive information, there is no guarantee that the maximum logit is the ground-truth one. Once we select the false-positive label as the correct one, our binary loss for updating the logits can avoid error accumulation to some extent. Besides, the average-based loss shows mediocre performance, since it aggregates the candidate logits without discrimination. Overall, the weight normalization strategy improves the robustness and generalization of the proposed method.

Sensitivity Analysis of Temperature Coefficient

The temperature coefficient is introduced to balance the weight contribution among labels. Figure 3 illustrates the performance of model under different settings. For Lost dataset, one can see that increasing λ leads a natural drop in accuracy. The reason is that the proposed weight normalization strategy gradually becomes the average-based one when $\lambda > 1$, of which the shortcomings have been analyzed above. For Kuzushiji dataset, the performance is relatively robust to this hyperparameter in most cases, and increasing λ properly will reduce variance to get a more stable performance but affect accuracy. In general, this strategy can be viewed as a generalization of those two classical strategies and fine-tuning λ can boost the model performance to some extent.

5 Conclusion

In this paper, we focused on the problem of learning with partially labeled data. Specifically, based on the PLL setting, we proposed a PLL binary classification loss and derived a novel estimator with estimation error bound. Furthermore, we proposed different logit adjustment methods for constructing the binary classifier, whose idea is directly encouraging $P_\theta(y \in s|\boldsymbol{x}) \to 1$. And we demonstrated the effectiveness of our algorithm in practice on both benchmark and real-world datasets. In most realistic cases, there is no guarantee that the ground-truth label of training examples must be one of their candidate labels, which motivates us to consider the robust partial loss, and we leave this for future work.

Acknowledgments

This work is supported by National Natural Science Foundation of China (61732006), National Key Research & Develop Plan(2018YFB1004401), National Natural Science Foundation of China (62072460, 62172424, 61772536, 62076245), and Beijing Natural Science Foundation (4212022).

References

[Briggs *et al.*, 2012] Forrest Briggs, Xiaoli Z Fern, and Raviv Raich. Rank-loss support instance machines for miml instance annotation. In *SIGKDD*, pages 534–542, 2012.

[Chou *et al.*, 2020] Yu-Ting Chou, Gang Niu, Hsuan-Tien Lin, and Masashi Sugiyama. Unbiased risk estimators can mislead: A case study of learning with complementary labels. In *ICML*, pages 1929–1938. PMLR, 2020.

[Clanuwat *et al.*, 2018] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.

[Cour *et al.*, 2009] Timothee Cour, Benjamin Sapp, Chris Jordan, and Ben Taskar. Learning from ambiguously labeled images. In *CVPR*, pages 919–926. IEEE, 2009.

[Cour *et al.*, 2011] T. Cour, B. Sapp, and B. Taskar. Learning from partial labels. *Journal of Machine Learning Research*, 12(4):1501–1536, 2011.

[Feng and An, 2018] Lei Feng and Bo An. Leveraging latent label distributions for partial label learning. In *IJCAI*, pages 2107–2113, 2018.

[Feng and An, 2019] Lei Feng and Bo An. Partial label learning by semantic difference maximization. In *IJCAI*, pages 2294–2300, 2019.

[Feng *et al.*, 2020] Lei Feng, Jiaqi Lv, Bo Han, Miao Xu, Gang Niu, Xin Geng, Bo An, and Masashi Sugiyama. Provably consistent partial-label learning. In *NeurIPS*, 2020.

[Grathwohl *et al.*, 2020] Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *ICLR*, 2020.

[Guillaumin *et al.*, 2010] Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. Multiple instance metric learning from automatically labeled bags of faces. In *ECCV*, pages 634–647. Springer, 2010.

[Hüllermeier and Beringer, 2006] Eyke Hüllermeier and Jürgen Beringer. Learning from ambiguously labeled examples. *Intelligent Data Analysis*, 10(5):419–439, 2006.

[Jin and Ghahramani, 2002] Rong Jin and Zoubin Ghahramani. Learning with multiple labels. In *NeurIPS*, volume 2, pages 897–904. Citeseer, 2002.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Liu and Dietterich, 2012] Liping Liu and Thomas G Dietterich. A conditional multinomial mixture model for superset label learning. In *Advances in neural information processing systems*, pages 548–556. Citeseer, 2012.

[Liu and Dietterich, 2014] Li-Ping Liu and Thomas G. Dietterich. Learnability of the superset label learning problem. In *ICML*, 2014.

[Lv *et al.*, 2020] Jiaqi Lv, Miao Xu, Lei Feng, Gang Niu, Xin Geng, and Masashi Sugiyama. Progressive identification of true labels for partial-label learning. In *ICML*, pages 6500–6510. PMLR, 2020.

[Mohri *et al.*, 2012] Mehryar Mohri, Afshin Rostamizadeh, and Ameet S. Talwalkar. Foundations of machine learning. In *Adaptive computation and machine learning*, 2012.

[Nguyen and Caruana, 2008] Nam Nguyen and Rich Caruana. Classification with partial labels. In *SIGKDD*, pages 551–559, 2008.

[Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

[Qian *et al.*, 2019] Qi Qian, Lei Shang, Baigui Sun, Juhua Hu, H. Li, and Rong Jin. Softtriple loss: Deep metric learning without triplet sampling. *ICCV*, pages 6449–6457, 2019.

[Seo and Huh, 2021] Junghoon Seo and Joon Suk Huh. On the power of deep but naive partial label learning. In *ICASSP*, pages 3820–3824. IEEE, 2021.

[Wen *et al.*, 2021] Hongwei Wen, Jingyi Cui, Hanyuan Hang, Jiabin Liu, Yisen Wang, and Zhouchen Lin. Leveraged weighted loss for partial label learning. In *ICML*, volume 139, pages 11091–11100. PMLR, 2021.

[Wu and Zhang, 2018] X. Wu and Min-Ling Zhang. Towards enabling binary decomposition for partial label learning. In *IJCAI*, 2018.

[Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[Yan and Guo, 2020] Yan Yan and Yuhong Guo. Partial label learning with batch label correction. In *AAAI*, volume 34, pages 6575–6582, 2020.

[Zeng *et al.*, 2013] Zinan Zeng, Shijie Xiao, Kui Jia, Tsung-Han Chan, Shenghua Gao, Dong Xu, and Yi Ma. Learning by associating ambiguously labeled images. In *CVPR*, pages 708–715, 2013.

[Zhang and Yu, 2015] Min-Ling Zhang and Fei Yu. Solving the partial label learning problem: An instance-based approach. In *IJCAI*, 2015.

[Zhang *et al.*, 2017] Min-Ling Zhang, Fei Yu, and Cai-Zhi Tang. Disambiguation-free partial label learning. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2155–2167, 2017.

[Zhang *et al.*, 2020] Yabin Zhang, Guang Yang, Suyun Zhao, Peng Ni, Hairong Lian, Hong Chen, and Cuiping Li. Partial label learning via generative adversarial nets. In *ECAI 2020*, pages 1674–1681. IOS Press, 2020.